CIRCUITOS DIGITAIS - PROF. VICTOR MIRANDA
ANÁLISES E EXERCÍCIOS COMENTADOS

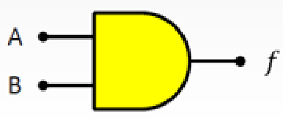
UNIDADE 3: PORTAS LÓGICAS

FUNÇÕES CUJA SAÍDA É DETERMINADA POR MEIO DO NÍVEL LÓGICO DE APENAS UMA VARIÁVEL, DENTRE AS DUAS:

Tabela verdade de $A \cdot B$

A	B	$A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

Porta Lógica AND:

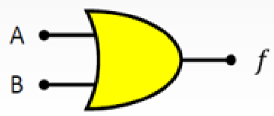


A saída será igual a 0 se ao menos uma das entradas for igual a 0 (independente da outra) e será igual a 1 somente se ambas as entradas forem iguais a 1 simultaneamente.

Tabela verdade de $A + B$

A	B	$A + B$
0	0	0
0	1	1
1	0	1
1	1	1

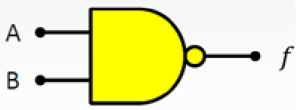
Porta Lógica OR:



A saída será igual a 1 se ao menos uma das entradas for igual a 1 (independente da outra) e será igual a 0 somente se ambas as entradas forem iguais a 0 simultaneamente.

Entradas		Saída
A	B	$\overline{A \cdot B}$
0	0	1
0	1	1
1	0	1
1	1	0

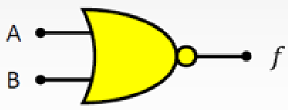
Porta Lógica NAND:



A saída será igual a 1 se ao menos uma das entradas for igual a 0 (independente da outra) e será igual a 0 somente se ambas as entradas forem iguais a 1 simultaneamente.

Entradas		Saída
A	B	$\overline{A + B}$
0	0	1
0	1	0
1	0	0
1	1	0

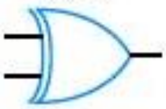
Porta Lógica NOR:



A saída será igual a 0 se ao menos uma das entradas for igual a 1 (independente da outra) e será igual a 1 somente se ambas as entradas forem iguais a 0 simultaneamente.

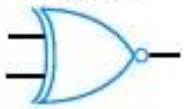
FUNÇÕES CUJA SAÍDA DEPENDE DOS NÍVEIS LÓGICOS DAS DUAS VARIÁVEIS:

XOR



x	y	F
0	0	0
0	1	1
1	0	1
1	1	0

XNOR



x	y	F
0	0	1
0	1	0
1	0	0
1	1	1

A XOR fornece 1 na saída quando as entradas forem diferentes entre si; e fornece 0 quando as entradas forem iguais entre si.

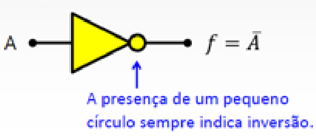
A XNOR fornece 1 na saída quando as entradas forem iguais entre si; e fornece 0 quando as entradas forem diferentes.

FUNÇÕES QUE DEPENDE DE APENAS UMA VARIÁVEL:

Tabela verdade de $\bar{A}$

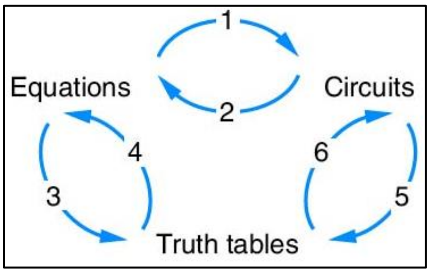
A	$\bar{A}$
0	1
1	0

Porta Lógica NOT:



Se a entrada for 0, a saída será 1;
Se a entrada for 1, a saída será 0.

UNIDADE 3: FORMAS DE REPRESENTAÇÃO DE UMA FUNÇÃO LÓGICA → & UNIDADE 5: REDES COMBINACIONAIS E MINIMIZACAO LOGICA VIA MAPAS DE KARNAUGH

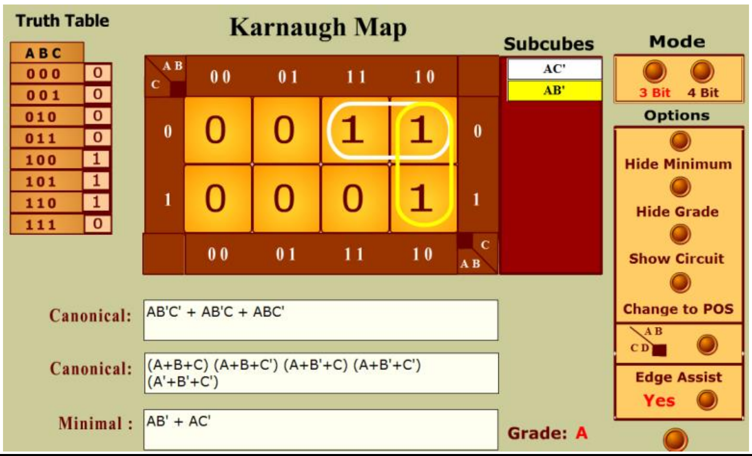


1) Derive as expressões na forma de SoP, PoS e Mínima:

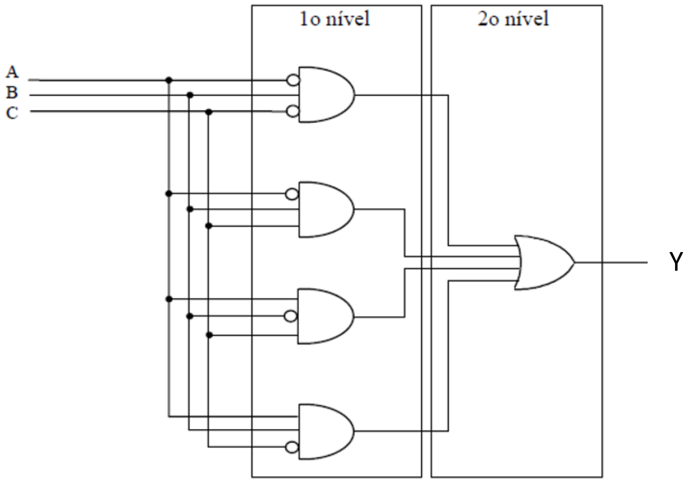
a) $\mathbf{R} = \overline{(\mathbf{A} \oplus \mathbf{B})\mathbf{C} + \overline{\mathbf{A}}}$ **com A (MSB) e C (LSB)**

No caso mais geral, monta a tabela verdade e preenche as saídas para cada linha de acordo com a função lógica enunciada → vide a execução deste passo na figura abaixo.

OBS*: Observe que o software utilizado para a obtenção da expressão mínima altera a exibição da ordem das 3 variáveis entre a tabela verdade e o mapa (o que não está errado se preenchido o mapa corretamente e condizente com a tabela verdade), porém, como dito em várias vezes em sala, o professor recomenda fortemente a construção do mapa de Karnaugh na mesma sequência de exibição das variáveis em relação à tabela verdade e esta sequência seria a mesma ordem de significância das variáveis.



b) No circuito a seguir: **A é o MSB e C, o LSB**



Y = A'BC' + A'BC + AB'C + ABC' = Y(SoP)

Esta expressão já está coincidentemente na forma canônica de SoP, onde cada termo-produto contém todas as variáveis (estando estas negadas ou não). Assim, podemos escrever a expressão anterior na forma:

$$Y(SoP) = \sum m(2, 3, 5, 6)$$

Busca fazer a conversão entre as formas canônicas sempre através da representação das expressões em índices decimais.

Se Y(SoP) = ∑ m(2, 3, 5, 6), então:

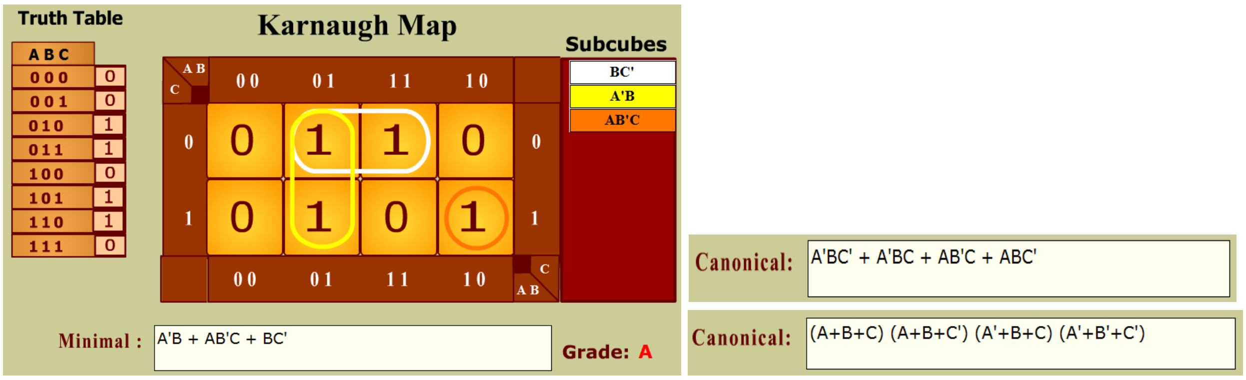
Y(PoS) = πM(0, 1, 4, 7)

A partir desta, escreve-se a forma canônica com base nas variáveis de entrada, **sempre atento à ordem de significância das variáveis, especialmente a mesma solicitada no enunciado**:

Y(PoS) = πM(0, 1, 4, 7)

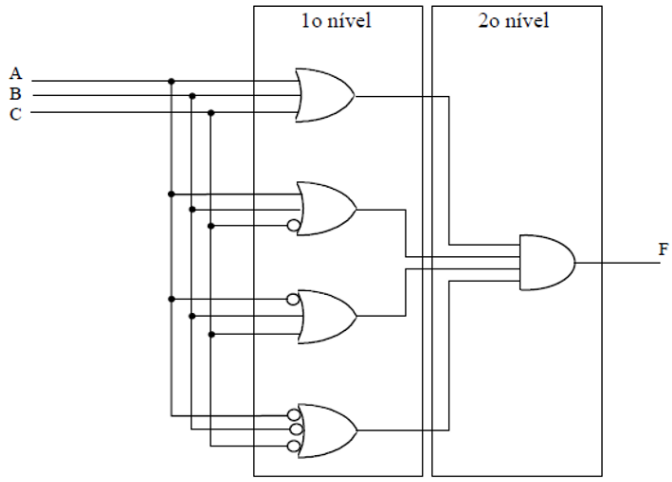
Y(PoS) = (A+B+C).(A+B+C').(A'+B+C).(A'+B'+C')

Por fim, a expressão mínima é obtida via Karnaugh* a seguir: vide observação OBS* acima.



---

c) No circuito a seguir: **C é o MSB e A, o LSB**



F = (C+B+A).(C'+B+A).(C+B+A').(C'+B'+A') = F(PoS)

Esta expressão já está coincidentemente na forma canônica de PoS, onde cada termo-soma contém todas as variáveis (estando estas negadas ou não). Assim, podemos escrever a expressão anterior na forma:

F(PoS) = πM (0, 4, 1, 7). Reordenando os índices na expressão, temos:

F(PoS) = πM (0, 1, 4, 7)

Busca fazer sempre a conversão entre as formas canônicas através da representação em índices decimais.

Se F(PoS) = πM (0, 1, 4, 7), então:

F(SoP) = ∑ m(2, 3, 5, 6)

A partir desta, escreve-se a forma canônica com base nas variáveis de entrada, **sempre atento à ordem de significância das variáveis, especialmente a mesma solicitada no enunciado**:

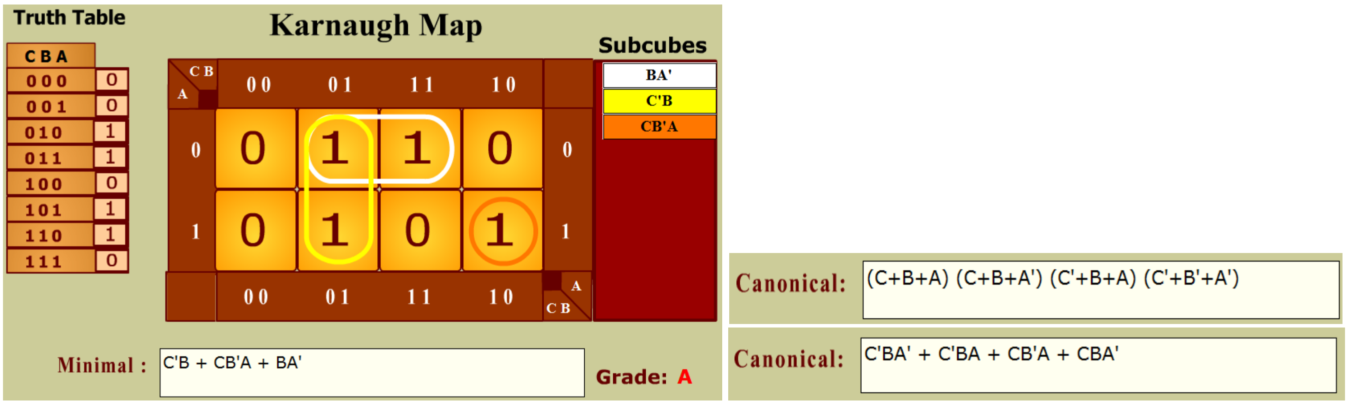
$$F(SoP) = \sum m(2, 3, 5, 6)$$

$$F(SoP) = C'BA' + C'BA + CB'A + CBA'$$

ATENÇÃO, devido à ordem de significância pedida ser CBA e não ABC, a expressão acima (resultado **correto**, conforme o enunciado) é diferente da expressão abaixo (resultado **incorreto**, conforme o enunciado – observe os termos diferentes em verde):

$$F(SoP) = A'BC' + A'BC + AB'C + ABC'$$

Por fim, a expressão mínima é obtida via Karnaugh* a seguir: vide observação OBS* acima.



---

d) S = X'Z'Y'W + Y'X com a seguinte ordem de significância: WXYZ

Esta expressão está quase na forma canônica de SoP. A diferença é que na forma canônica SoP, cada termo-produto deve conter todas as variáveis (estando estas negadas ou não).
Com isso, usa-se, nesse caso, o seguinte artifício:

1) Acrescentam-se as combinações possíveis das variáveis que faltam para completar o termo.
Para N variáves faltantes no termo, são possíveis 2^N combinações entre elas e, portanto, serão acrescentados 2^N termos à expressão. No nosso exemplo, faltam 2 variáveis em um dos termos-produtos e são acrescentados 2^2 = 4 termos (em vermelho).

$$S = X'Z'Y'W + Y'X\,(Z'W') + Y'X\,(Z'W) + Y'X\,(ZW') + Y'X\,(ZW)$$

2) Agora temos que cada termo-produto contém todas as variáveis (barradas ou não) → forma canônica SoP. Porém, faz-se necessário reordenar, na sequência, as variáveis em cada termo-produto de acordo com a ordem de significância.

$$S(SoP) = WX'Y'Z' + W'XY'Z' + WXY'Z' + W'XY'Z + WXY'Z$$

3) A cada termo-produto, associa-se o respectivo mintermo.

$$S(SoP) = \sum m(8, 4, 12, 5, 13)$$

$$S(SoP) = \sum m(4, 5, 8, 12, 13)$$

4) Busca fazer sempre a conversão entre as formas canônicas através da representação em índices decimais.

Se $S(SoP) = \sum m(8,4,12,5,13) = \sum m(4,5,8,12,13)$, então:

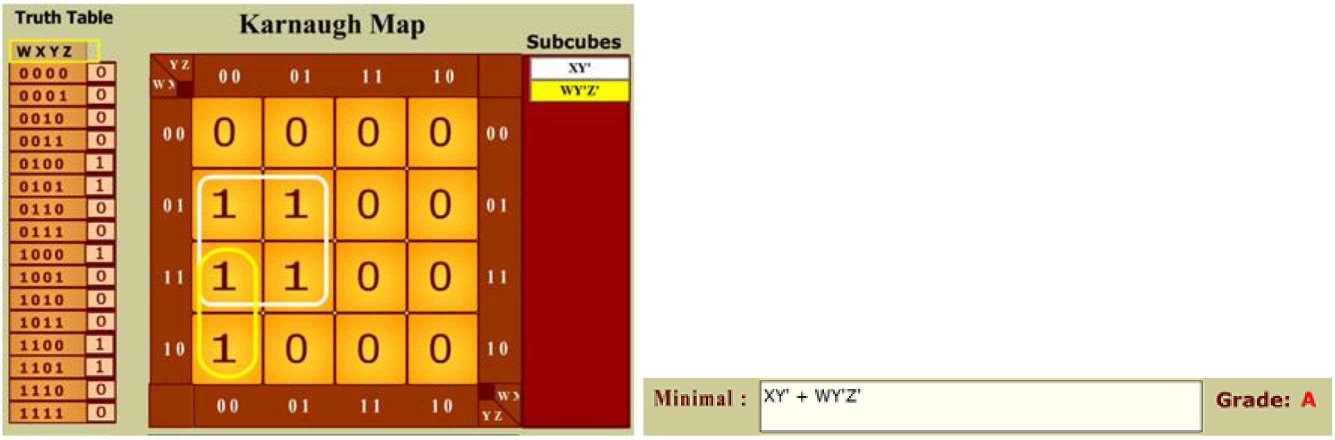
$$S(PoS) = \prod M(0,1,2,3,6,7,9,10,11,14,15)$$

A partir desta, escreve-se a forma canônica com base nas variáveis de entrada, sempre atento à ordem de significância das variáveis:

$$S(PoS) = \prod M(0,1,2,3,6,7,9,10,11,14,15)$$

$$S(PoS) = (W+X+Y+Z).(W+X+Y+Z').(W+X+Y'+Z).(W+X+Y'+Z').(W+X'+Y'+Z).(W+X'+Y'+Z').(W'+X+Y+Z').(W'+X+Y'+Z).$$
$$(W'+X+Y'+Z').(W'+X'+Y'+Z).(W'+X'+Y'+Z')$$

A expressão mínima é obtida via Mapa de Karnaugh a seguir:



UNIDADE 6: PROJETO LÓGICO COMBINACIONAL

[NN95, ex. 2.22] Um alarme é projetado para monitorar quatro sinais de entrada. O sinal A vem da chave de controle secreta. O sinal B vem de um sensor de pressão debaixo de um cofre de aço em um armário trancado. O sinal C vem de um relógio movido a bateria. O sinal D está ligado a uma chave na porta de um armário trancado. As seguintes condições produzem uma tensão de nível ALTO em cada linha:

- A chave de controle está acionada.
- O cofre está em sua posição normal no armário.
- O relógio está entre 09:00 e 15:00 horas (horário bancário).
- A porta do armário está fechada.

Escreva a expressão booleana para o alarme que produz uma lógica 1 (tocar um alarme) quando o cofre é movido e a chave de controle é acionada, ou quando o armário é aberto após o expediente bancário, ou quando o armário é aberto com a chave de controle desligada.

<u>Etapas de um Projeto Lógico:</u>

1) **<u>Interpretação Lógica:</u>** consiste em identificar a(s) variável(is) de entrada e de saída e atribuir níveis lógicos a cada uma delas de acordo com cada situação enunciada.
Geralmente esta atribuição é livre e de escolha do projetista, porém neste exemplo o enunciado amarra as atribuições dos níveis lógicos para cada variável:

Variáveis de Entrada:
A = Acionamento da chave de controle
1 = chave de controle é acionada
0 = chave de controle não é acionada

B = Posição do cofre no armário
1 = o cofre está em sua posição normal no armário
0 = o cofre é movido de sua posição normal no armário

C = Marcação do horário do relógio
1 = relógio marca horário bancário (no caso, entre 09:00 hs e 15:00 hs)
0 = fora do horário bancário (no caso, antes de 09:00 hs e após 15:00 hs)

D = Situação da porta do armário

1 = a porta do armário está fechada
0 = a porta do armário está aberta

Variável de Saída:
S = Alarme
1 = acionado
0 = não acionado

2) **<u>Tabela Verdade:</u>** uma vez identificada a quantidade de variáveis, cria-se a tabela verdade.

O preenchimento das saídas para cada linha da tabela, ou seja, para cada combinação das variáveis entradas, se dá verificando se cada combinação (linha) atende às premissas enunciadas pelo problema.

No caso, **o alarme será acionado (NL 1)** se **pelo menos uma (OU)** das condições abaixo for atendida:

Condição 1: o cofre é movido (B=0) E chave de controle é acionada (A=1)

OU

Condição 2: quando o armário é aberto (D=0) E o relógio marca horário fora do expediente bancário (C=0)

OU

Condição 3: quando o armário é aberto (D=0) E a chave de controle está desligada (A=0)

Observe onde cada condição é atendida:

Condição 1: atendida pelas linhas 8, 9, 10 e 11

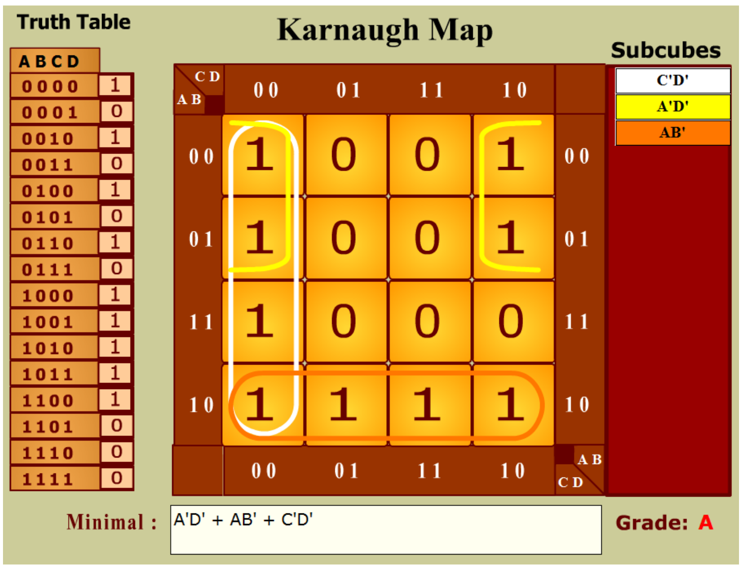
Condição 2: atendida pelas linhas 0, 4, 8 e 12

Condição 3: atendida pelas linhas 0, 2, 4 e 6

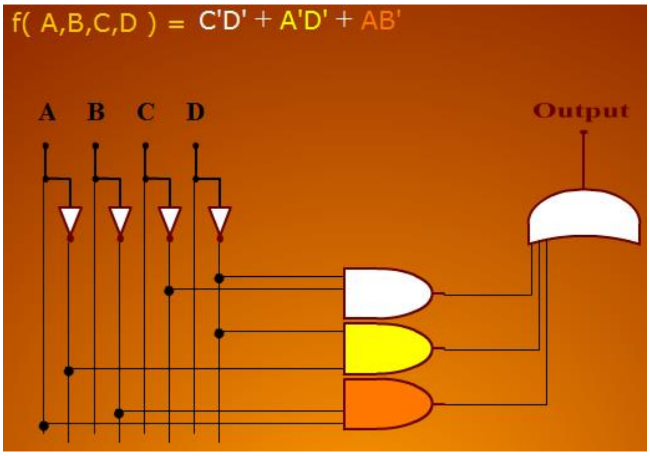
Observe que algumas linhas atendem a mais de uma condição ao mesmo tempo. As demais linha da tabela verdade não citadas acima não atendem a nenhuma condição e geram saída em 0.

OBS: Vide a tabela verdade na figura abaixo.

3) **<u>Expressão Mínima via Mapa de Karnaugh (K-Map):</u>** uma vez preenchida a tabela verdade, constrói-se o mapa de Karnaugh e deriva-se a expressão mínima.



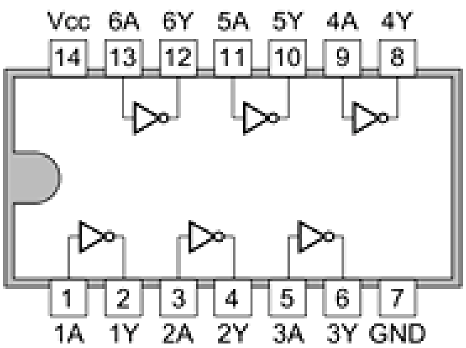
4) **Circuito Mínimo:** uma vez obtida a expressão mínima, constrói-se o circuito lógico correspondente a esta.



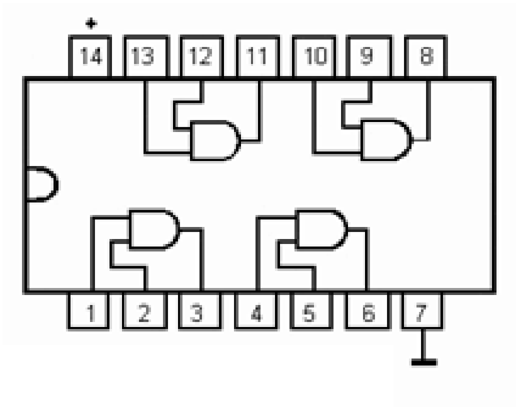
ASPECTOS PRÁTICOS DE LABORATÓRIO:

Circuitos Integrados (CI's ou Chip's) utilizados nas aulas práticas:

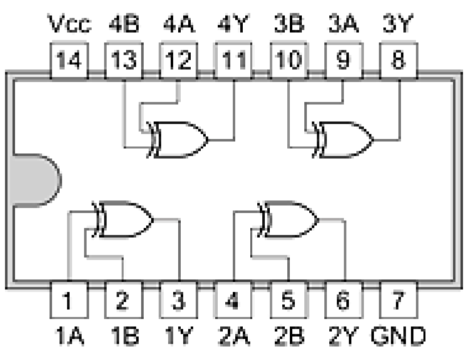
Portas NOT (CI 7404)



Portas AND (CI 7408)



Portas XOR (CI 7486)



Portas OR (CI 7432)